Introduction to AFS
IMSA Intersession 2003

AFS Servers and Clients

Brian Sebby, IMSA '96

AFS Server processes

- Several processes are run to manage the various AFS servers. These are located in /usr/afs/bin/.
- The configuration files for the AFS servers are located in /usr/afs/etc/.
- Local configuration files for the particular AFS server are stored in /usr/afs/local/.
- The databases (authentication, protection, volume location, and backup) are located in /usr/afs/db/. These are stored on disk as regular files.
- The file server serves volumes out of its vice partitions which are mounted as /vicep<a-z>. These partitions use the native file system format to store their data. For example, under Linux this is usually ext2 or ext3, and under Solaris this is ufs.
- Before any server processes can be run, AFS extensions need to be loaded into the kernel.

The Basic OverSeer Server

- The Basic OverSeer Server does not directly provide any AFS services. It is an administrative process that controls the starting, stopping, and restarting of the other AFS server processes, can run AFS cron jobs, and monitor the status of the other server processes.
- The server process is called `bosserver`, and you interact with the `bosserver` through the `bos` command suite.
- It knows what processes it is running or monitoring from a configuration file called `BosConfig`.
- It can restart the AFS server processes if they fail, or restart them on a regular schedule to ensure that they are working.
- It is usually started in the AFS initialization script.
- It does not monitor or restart itself.

The Authentication Server

- The Authentication Server maintains the authentication database for AFS.
- It verifies a user’s identity and issues tokens for AFS access.
- It allows the various servers to authenticate to each other.
- It stores the AFS server encryption key, which is shared by all database servers.
- The server process is called `kaserver`, and you interact with the `kaserver` through the `kas` command suite.

The Protection, Volume Location, and Backup Servers

- The Protection Server maintains the protection database, which contains such information as user names, group names, UIDs, and GIDs.
- It gives this information to the fileserver when information about a user or group is needed.
- The server process is called `ptserver`, and you interact with the `ptserver` though the `pts` command suite.
- The Volume Location Server maintains the VLDB and tells the cache manager where to find volumes on disk that it is trying to access.
- The server process is called `vlserver`, and you interact with the `vlserver` through the `vos` command suite.
- The Backup server maintains the backup database. Its server process is called `buserver`, and you interact with it through the `backup` command suite. We will not be covering backups in this intersession.

The File Server

- The File Server processes manage the storing and delivering of AFS data.
- Three processes make up the file server: `fileserver`, `volserver`, and `salvager`. The `bosserver` treats these as one unit.
- The `fileserver` process stores files in volumes in the AFS partitions. It also stores ACLs and delivers files if the user has permission to access that data. The `fs` command suite interacts with `fileserver`.
- The `volserver` process handles data access on a volume level, such volume creation, deletion, and moving. The `vos` command suite interacts with `volserver`. Note that this is not the same process as `vlserver`, which manages the VLDB.
- The `salvager` is essentially the AFS version of `fsck`. It can fix errors in the file system when needed (if, say, the file server crashes in the middle of an update.) It is usually run automatically by the `bosserver` when needed, but can be manually run by using the `salvage` command.

AFS Configuration Files - `CellServDB`

- There are a few files stored in /usr/afs/etc/ that are used to configure the servers.
- The `CellServDB` file lists the database servers for your cell, and any foreign cells you wish to contact. It uses the server IP addresses instead of their host names. You can have only one IP address listed per database server.
- The format of `CellServDB` is as follows:

```
>cellname       # Longer version of cell name
DB server IP address1 #servername
DB server IP address2 #servername
```

- For instance, the cell we are creating for this intersession will have a `CellServDB` file like this:

```
>imsa.edu       # Ill. Math and Science Academy
10.10.10.1      #imsasun.imsa.edu
10.10.10.2      #einstein.imsa.edu
10.10.10.4      #mudgate.imsa.edu
```

AFS Configuration Files – UserList, KeyFile, and ThisCell

- The `UserList` file lists the users who can execute privileged `bos` and `vos` commands.
- The system administrators of an AFS cell should have separate admin accounts that are members of this list.
- The `bos adduser` command adds a user to the list, and `bos listusers` lists the members of the list.
- The `KeyFile` file stores server encryption keys. It should be the same on all AFS servers to allow them to decrypt tickets from AFS clients. There should be one key that matches the key for the AFS entity “afs” in the authentication database.
- The `ThisCell` file is a small text file that contains the name of the local AFS cell. In our case the file would simply contain the string “imsa.edu”.

AFS Configuration Files - `BosConfig`

- The `BosConfig` file is the configuration file for bosserver, and is stored in /usr/afs/local/. `BosConfig` can be different for each AFS server.
- The `BosConfig` can control three types of processes: simple, fs, and cron.
- The simple processes are ones like `vlserver` that are just a single command that needs to be run.
- An fs process is a complex process that involves multiple commands. The trio of file server processes use this type of command.
- The cron processes are commands that are run at a certain time. This can include things like releasing volumes, creating backup volumes, etc.
- The `BosConfig` file should never be edited by hand – you should always use the bos command suite to manage it.

AFS Clients

- AFS clients are responsible for communicating with the AFS servers and serving AFS data to users.
- Clients have a local cache that is used while the data is being accessed.
- The main client process is called `afsd`. It mounts the AFS file system in the correct place and loads the configuration data about the cell into the kernel memory. It also sets up the cache.
- The `afsd` process is normally run when the client is booted from an AFS rc script. An AFS rc script will also usually load the AFS kernel extensions as well.

AFS Client Configuration

- Configuration files for AFS clients are located in /usr/vice/etc/. The cache is located in /usr/vice/cache/.
- The main configuration file is called `cacheinfo`. It is of the format:
  ```
  <AFS mount point>:<cache location>:<cache size>
  ```
  In our case, I have set aside a 128M partition for the cache, so our cacheinfo file would look like:
  ```
  /afs:/usr/vice/cache:100000
  ```
- It is recommended that you use a separate partition for the cache, and configure the cache to use 80% of the disk space. AFS will use an additional 10% of the space, and most operating systems reserve 10% of the drive for system use.
- Other files in the directory include `CellServDB` and `ThisCell`. These have the same format as the files in the server configuration directories, and usually have the same contents as the ones in those directories.
- AFS servers usually run an AFS client, but it is not necessary to do so.

Installing AFS Servers and Clients

- The procedures for installing servers and clients varies depending on the type of machine you are installing on.
- Please refer to the *AFS Quick Beginnings* guide in the AFS documentation collection at http://www.openafs.org/doc/index.htm
- That document provides a step-by-step guide to installing AFS depending on the operating system you are using.
- The general series of steps you must follow for all platforms is as follows: load the AFS extensions into memory, start the bosserver without authentication, create entries for the various server processes, add entries for afs and the admin account to the authentication database, create the root.afs and root.cell volumes, and restart the bosserver with authentication enabled.
- We will discuss managing the AFS server processes in more detail later.